

1     **SYSTEM AND METHOD FOR DECODING AND ANALYZING BARCODES USING**  
2                             **A MOBILE DEVICE**

3  
4     **FIELD OF THE INVENTION**

5             The present invention relates generally to the field  
6 of barcode scanning and decoding using a mobile device.  
7 More specifically, the present invention uses optical  
8 decoding techniques to decode barcodes acquired via a  
9 camera phone or other similarly equipped device.

10  
11    **PARENT CASE TEXT**

12            This application claims the benefit of provisional  
13 application No. 60/487,237 filed 07/16/2003.

14  
15    **BACKGROUND OF THE INVENTION**

16            Barcodes have been utilized for identifying and  
17 pricing objects for more than thirty years. Most  
18 typically, barcodes are used in retail to identify the item  
19 of merchandise. For example, a gallon of milk may contain  
20 a barcode that, when scanned, will notify the cashier of  
21 the price of the milk.

22            Yet in recent years, barcodes have acquired new  
23 purposes as computers and barcode scanners have become more  
24 portable. The circuitry required to scan a conventional

1 one-dimensional barcode can now be housed in a device as  
2 small as a typical keychain. As a result, many mobile  
3 telephones, personal digital assistants ("PDAs"), and  
4 pagers can be retrofitted with or connected to a laser-  
5 based scanning device. This allows the mobile device to  
6 function as a scanner capable of storing hundreds or  
7 thousands of scanned barcodes.

8 Mobile devices with attached scanners have allowed for  
9 the development of a new niche in the wireless electronics  
10 business. Some companies have developed software and  
11 hardware which allows a user to scan any barcode and be  
12 redirected to media information (e.g., a website, product  
13 description, price, etc.) about the scanned product. These  
14 programs provide a link between the physical and online  
15 world which previously did not exist.

16 However, mobile devices with attached scanners possess  
17 some drawbacks which have curtailed their expansion into  
18 the mobile marketplace. First, there are few mobile  
19 devices produced for the general public that contain  
20 integrated laser-based scanners. Therefore, for a user to  
21 acquire scanning capability for a mobile device, he/she  
22 must purchase additional equipment. The additional  
23 scanning equipment also adds size and weight to the mobile  
24 device, thereby reducing its mobility.

1       Currently, many cell phones and mobile devices are  
2   available with built-in cameras. The explosion of the  
3   availability of affordable digital cameras and their  
4   inclusion into mobile devices is driven by several factors.  
5   One of the most important is the recent availability of  
6   inexpensive image sensors based on CMOS technology. The  
7   cameras on these devices provide a means for capturing the  
8   barcode information which was previously only accessible  
9   via a laser-based scanner. Decoding barcode images from  
10   digital cameras included in mobile devices presents several  
11   difficult problems. These problems go well beyond the  
12   challenges addressed in commercial barcode readers.  
13   Barcode decoding algorithms from commercial products will  
14   not consistently decode images from a consumer portable  
15   device. Some of these problems are addressed below:

16   ***Lighting:***

17       Most mobile devices with integrated digital cameras do  
18   not have built-in flashes and rely solely on the ambient  
19   light for illumination. Using highly variable ambient  
20   light makes pattern recognition much more difficult.  
21   Shadows, shading across the length of a barcode,  
22   overexposure, underexposure, and similar problems that are  
23   typical of any camera not utilizing a flash can foil

1 traditional barcode decoding algorithms that are designed  
2 for highly controlled lighting environments.

3 **Size:**

4 The distance between a digital camera and its target  
5 object is not usually rigidly controlled. This translates  
6 into a large range of possible sizes (magnifications) that  
7 a barcode can have on a fixed size image sensor.

8 **Skew:**

9 As any photographer knows, taking pictures at an angle  
10 changes the apparent shape of the object to a viewer. A  
11 barcode with a rectangular shape, when viewed straight-on,  
12 can look like a trapezoid (or irregular quadrilateral) when  
13 viewed from an angle. The location and addressing of image  
14 pixels for a barcode change dramatically when viewed from  
15 the side, or tilted. Algorithms to decode barcodes from  
16 digital images must be able to address images distorted  
17 from skewed viewing angles, but must do so within the  
18 constraints of limited hardware, processing power, and  
19 memory typically found in mobile devices such as PDAs and  
20 handsets.

21 **Battery power:**

22 Portable devices run on batteries - the smaller the  
23 better. Barcode decoding algorithms for cameras must be  
24 very efficient so as to use low amounts of CPU power.

1 Charge coupled diode ("CCD") devices and barcode scanners  
2 using laser light generally require a large amount of  
3 power, and are not well suited for battery powered,  
4 handheld devices.

5 ***Color Imagers:***

6 Consumer oriented devices such as mobile handsets  
7 generally are designed with color image sensors. However,  
8 barcode scanning typically operates best with gray-scale  
9 information. Color data typically requires three times the  
10 amount of storage and handling required by gray-scale.  
11 Data needs to be transferred through the camera's CPU and  
12 memory to be processed. For color imagers, specific image  
13 processing algorithms are required in order to avoid  
14 problematic image artifacts during the translation from  
15 color to grayscale.

16 ***Focus:***

17 Digital cameras for portable devices are usually  
18 designed to work at a variety of distances. The need for a  
19 wider range of focus in cameras results in a trade off  
20 between the cost of the lens component and the sharpness of  
21 a typical image. Decoding algorithms for embedded digital  
22 cameras must be able to cope with a moderate degree of  
23 focus problems.

24 ***Low-cost lens components:***

1        In order to meet cost constraints of many portable  
2 device markets, manufacturers often compromise on the  
3 optical quality of camera lenses. This can present decoding  
4 technology with a different set of challenges from the  
5 simple focal length based focus problem noted above. Low-  
6 cost lens components can produce image distortions that are  
7 localized to a specific region or form a changing gradient  
8 across the image. This requires additional sophistication  
9 for decoding algorithms.

10    **Limited resolution:**

11        The cost of a digital imaging CMOS sensor increases as  
12 the number of image pixels increases. Although the Asian  
13 market has seen the release of general purpose consumer  
14 devices like PDAs and cell phones with "megapixel" image  
15 resolution, it is unlikely these devices will be released  
16 in the mainstream European and North American markets in  
17 the near future. With fewer pixels to work with, it is  
18 significantly more difficult to reliably decode barcodes  
19 from images.

20        Based on the aforementioned described problems with  
21 mobile digital imaging, there clearly exists a need for a  
22 system capable of capturing, decoding, and analyzing  
23 barcode information obtained from a digital camera enabled  
24 mobile device. Such a system would enable the average

1 mobile device user to accurately and reliably scan and  
2 decode any barcode available.

3

#### 4 **SUMMARY OF THE INVENTION**

5       The present invention provides a software application  
6 and system (hereinafter referred to as "ScanZoom") designed  
7 to successfully process and decode barcodes acquired via  
8 digital imaging techniques. ScanZoom software empowers a  
9 user to use a cell-phone or PDA equipped with a digital  
10 camera as a scanner of barcodes (one-dimensional and two-  
11 dimensional) or any other similar machine-readable code. It  
12 seamlessly integrates the barcode scanning technology with  
13 the digital camera (built-in or attached) of the cell  
14 phones /PDAs/Pocket PCs.

15       To utilize the ScanZoom software, a user downloads the  
16 ScanZoom onto his/her cell phone or PDA through wireless  
17 access protocol ("WAP"), infrared, or Bluetooth®  
18 connectivity. However, any protocol which allows a user to  
19 download a program to a mobile device may be utilized to  
20 download ScanZoom. Once the user has downloaded ScanZoom,  
21 the user launches the application. This causes ScanZoom to  
22 properly initialize the digital camera coupled to the  
23 mobile device to accept input. It starts the digital  
24 camera by calling its Application identification.

1       The user then takes a picture of the barcode using the  
2 digital camera. As soon as the barcode is captured, the  
3 ScanZoom software decodes the barcode utilizing a decoding  
4 engine integral to the ScanZoom software. Alternatively,  
5 the ScanZoom software may send the image of the barcode to  
6 a central server for decoding by sending a multimedia  
7 message service ("MMS") message to the server containing  
8 the barcode image.

9       If ScanZoom sends a MMS message to the server, it  
10 launches the MMS Application Id, composes the appropriate  
11 message on the fly, and then sends the message to the  
12 SMS/MMS server. On the server side, a global system for  
13 mobile communications ("GSM") modem connected to the server  
14 has the appropriate security identity module ("SIM") card  
15 and takes the services from any mobile service provider.  
16 The server fetches the MMS message from the GSM modem queue  
17 and performs appropriate action depending upon the message.  
18 The server can then send back a simple SMS message or can  
19 send back a multimedia message service ("MMS") message  
20 which can launch a WAP browser on the mobile device and  
21 direct it to the appropriate website, or send back  
22 information to the user through any other existing wireless  
23 protocol.



1       The location of the decoding depends entirely upon the  
2   processing capabilities of the mobile device utilized. For  
3   example, if the ScanZoom software is operating on a mobile  
4   device with lower system capabilities, such as a first  
5   generation camera phone, the mobile device will  
6   automatically send the digital image of the barcode to a  
7   server for decoding.

8       The barcode decoding engine continuously runs in a  
9   loop until it's able to decode the image taken by the  
10   digital camera into a barcode. If the barcode cannot be  
11   properly resolved, the user is prompted to take another  
12   picture of the desired barcode.

13       Additionally, the barcode decoding may also be  
14   performed in real time. The ScanZoom software accomplishes  
15   this by constantly capturing and processing the image of  
16   the barcode until it is correctly decoded. This eliminates  
17   the extra step of the user having to press a button to  
18   capture an image of the barcode.

19       After the barcode has been correctly resolved either  
20   by the mobile device or the server, the mobile device  
21   displays the appropriate media content to the user. The  
22   media content displayed to the user depends entirely on the  
23   barcode scanned. For example, if a user scans a barcode on  
24   a compact disc, the ScanZoom application may launch a WAP

1 browser and direct the user to a site which allows the user  
2 to purchase the compact disc electronically. As another  
3 example, if a user scans a barcode located on a food item,  
4 the server may return a SMS message to the mobile device  
5 indicating the nutritional contents of the scanned item.

6 Therefore, it is an object of the present invention to  
7 provide a software application and system capable of  
8 accurately and reliably decoding barcodes and other  
9 machine-readable codes acquired via a digital camera  
10 connected to a mobile device.

11 Another object of the present invention is to provide  
12 a software application and system which allows for the  
13 decoding of barcodes in a wide range of conditions.

14 An additional object of the present invention is to  
15 provide a software application and system for decoding  
16 barcodes which is quick and responsive.

17 Yet another object of the present invention is to  
18 provide a software application and system for decoding  
19 barcodes which is robust under adverse lighting, imaging,  
20 and focusing conditions.

21 Still another object of the present invention is to  
22 provide a software application and system for decoding  
23 multiple barcode formats.

1        Another object of the present invention is to provide  
2        a software application and system which does not adversely  
3        affect device performance, usability, or form factor.

4        Furthermore, an object of the present invention is to  
5        provide a software application and system for decoding  
6        barcodes which does not significantly impact device power  
7        consumption nor degrade general camera performance.

8        It is another object of the present invention to  
9        provide a barcode decoding system which requires minimal or  
10       no changes to the manufacturing process of the mobile  
11       devices.

12       An additional object of the present invention is to  
13       provide a barcode decoding system having a low incremental  
14       cost per device.

15       Another object of the present invention is to provide  
16       a highly reliable barcode decoding system requiring minimal  
17       user support.

18       These and other objects of the present will be made  
19       clearer with reference to the following detailed  
20       description and accompanying drawings.

21

22       **BRIEF DESCRIPTION OF THE DRAWINGS**

1        FIG. 1 depicts a schematic diagram of the network  
2        configuration utilized in the preferred embodiment of the  
3        invention.

4        FIG. 2A depicts a flowchart showing the steps the  
5        ScanZoom software utilizes to image and decode a barcode.

6        FIG. 2B depicts a flowchart that the ScanZoom software  
7        utilizes if MMS messaging is utilized to transmit the  
8        captured barcode.

9        FIG. 2C depicts a flowchart that the ScanZoom software  
10        utilizes if WAP messaging is utilized to transmit the  
11        scanned barcode.

12       FIG. 3A depicts a schematic diagram showing the  
13       product architecture of the ScanZoom software application.

14       FIG. 3B depicts a flowchart showing the steps utilized  
15       by ScanZoom to acquire the barcode image and prepare it for  
16       decoding.

17       FIG. 4A depicts a flowchart showing the process  
18       utilized by the decoding engine to enhance an image before  
19       decoding.

20       FIG. 4B depicts a flowchart showing the process  
21       utilized by the decoding engine to decode a barcode.

22       FIG. 5A depicts a flowchart showing the process  
23       utilized by the ScanZoom software to sharpen an image.

1        FIG. 5B depicts a typical barcode image acquired using  
2        a digital camera.

3        FIG. 5C depicts the barcode of FIG. 5B after it has  
4        undergone sharpening utilizing the sharpening filter  
5        depicted in FIG. 5A.

6

7        **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)**

8        The following presents a detailed description of a  
9        preferred embodiment (as well as some alternative  
10        embodiments) of the present invention. However, it should  
11        be apparent to one skilled in the art that the described  
12        embodiment may be modified in form and content to be  
13        optimized for a wide variety of situations.

14        With reference first to FIG. 1, shown is a schematic  
15        diagram of the network configuration utilized in the  
16        preferred embodiment of the present invention. In this  
17        figure, product 101 contains barcode 103 which may be  
18        placed on product 101 in a variety of ways. For example,  
19        barcode 103 may be printed directly onto product 101  
20        utilizing conventional printing techniques. Alternatively,  
21        barcode 103 may be affixed to product 101 utilizing a  
22        sticker, tag, etc.

23        Barcode 103 may be any machine-readable code utilizing  
24        either a public standard encoding symbology or a

1 proprietary symbology. Some examples of one and two  
2 dimensional symbologies include, but are not limited to,  
3 UPC-A, UPC-E, ISBN, RSS-14, RSS-14E, RSS-14L, Interleaved 2  
4 of 5, EAN/JAN-8, EAN/JAN-13, Code 39, Code 39 Full ASCII,  
5 Code 128, PDF417, QR Code, Data Matrix, and Optical  
6 Intelligence 2D.

7 To scan barcode 103, a user utilizes mobile device 105  
8 with attached or embedded digital camera 107. The user  
9 first launches the ScanZoom application on mobile device  
10 105. If the ScanZoom software has not yet been loaded onto  
11 mobile device 105, a user may do so by downloading the  
12 program via WAP, Bluetooth, or infrared connectivity.

13 Mobile device 105 may be any device capable of  
14 digitally imaging barcode 103 such as a camera phone,  
15 mobile phone with camera attachment, PDA, PDA with camera  
16 attachment, Pocket PC, Palm device, laptop, desktop, etc.

17 Once an image of barcode 103 has been acquired by  
18 mobile device 105, the ScanZoom software loaded on mobile  
19 device 105 decodes the barcode directly utilizing mobile  
20 device 105's internal circuitry. The ScanZoom software can  
21 be ported to most any mobile device operating system  
22 including, but not limited to, Symbian OS, Palm OS, Windows  
23 CE, Windows Mobile, and SmartPhone. The decoded barcode  
24 information is then sent to wireless network 109 via

1 SMS/MMS message 111. Wireless network 109 routes SMS/MMS  
2 message 111 to the appropriate server 113. Server 113 then  
3 processes the received barcode information and relays the  
4 appropriate media content 115 to mobile device 105 via  
5 wireless network 109.

6 Alternatively, mobile device 105 may not process the  
7 barcode image at all. In this situation, mobile device 105  
8 would send the entire barcode image to server 113 via MMS  
9 message 111. ScanZoom software loaded onto server 113  
10 would be responsible for correctly decoding barcode 103.

11 Yet in another embodiment, mobile device may scan  
12 barcode 103 in real time, thereby eliminating the step of  
13 capturing a photo of the barcode. In this embodiment, the  
14 ScanZoom software located on mobile device 105 constantly  
15 acquires images of barcode 103 and stores them in memory.  
16 Each of these images is subsequently processed until the  
17 barcode is correctly decoded. Once barcode 103 has been  
18 decoded, mobile device 105 stops acquiring images of  
19 barcode 103.

20 Server 113 may process SMS/MMS message 111 in many  
21 different ways. In a first embodiment, server 113 may use  
22 relational database 114 to pull up product information  
23 pertaining to product 101. The server would then forward  
24 the product information to mobile device 105 via a SMS/MMS

1 message. If a MMS message is sent to mobile device 105,  
2 this may cause a WAP browser loaded on mobile device 105 to  
3 be redirected to the appropriate site.

4 In a second embodiment, server 113 may process SMS/MMS  
5 message 111 by using relational database 114 to pull up  
6 product information related to product 101 and then utilize  
7 search engine 117 to search for similar products or  
8 information pertaining to it. The results of the search  
9 would then be sent to mobile device 105 via a SMS/MMS  
10 message. As should be evident from these two described  
11 embodiments, the barcode information can be utilized in an  
12 almost limitless amount of ways.

13 To provide additional security, a user of mobile  
14 device 105 may be prompted to provide a username and/or  
15 password to access server 113. Server 113 would use user  
16 database 119 to properly authenticate users. Users not  
17 having an account contained in user database 119 would not  
18 be granted access to server 113 in any capacity.

19 Referring next to FIG. 2A, shown is the process  
20 utilized by ScanZoom to acquire and decode barcode 103 once  
21 the application has been launched. A user first launches  
22 the ScanZoom program on mobile device 105 by selecting its  
23 icon from a list of programs located on the device in step  
24 201. Generally, this will initialize the camera and all



1 the network resources needed by ScanZoom for the  
2 application to function properly. Once the user has  
3 started the program, a message appears directing the user  
4 to take a picture of barcode 103 in step 203. ScanZoom  
5 provides the user with a "preview" window which functions  
6 as a viewfinder, allowing the user to properly center and  
7 align the barcode before taking a picture. Once the  
8 barcode image has been acquired by digital camera 107,  
9 ScanZoom attempts to decode the barcode utilizing its  
10 proprietary barcode decoding algorithm in step 205. If the  
11 software cannot decode barcode 103 on a first attempt in  
12 step 205, the ScanZoom software attempts to decode the  
13 image a finite amount of times. The number of decoding  
14 attempts can either be fixed or able to be altered by the  
15 user. Once barcode 103 has been properly decoded as  
16 verified in step 207, mobile device 105 prepares the  
17 decoded information for transfer to server 113 to undergo  
18 further processing in step 209.

19 FIG. 2B depicts the process utilized by the ScanZoom  
20 software if the decoded barcode information is sent to  
21 server 113 via SMS/MMS message 111. In this scenario,  
22 mobile device 105 first encodes the decoded barcode  
23 information into SMS/MMS message 111 in step 221. SMS/MMS  
24 message 111 is then processed by server 113 in step 223.

1 As previously discussed with respect to FIG. 1, server 113  
2 may utilize any combination of relational database 114,  
3 search engine 117, or user database 119 to process the  
4 barcode information. Once the barcode has been processed  
5 in step 223, server 113 sends the resulting media content  
6 115 back to mobile device 105 in step 225. Media content  
7 115 is then displayed on the screen of mobile device 105 in  
8 step 227.

9 Alternatively, if a WAP connection is utilized to send  
10 the barcode information to server 113 via wireless network  
11 109, the process depicted in FIG. 2C is utilized. First,  
12 mobile device 105 conditions the barcode information for  
13 transfer via WAP in step 241. Once the barcode information  
14 arrives at server 113, the barcode information is  
15 depacketized and processed by server 113 in step 243. Once  
16 the barcode has been processed in step 243, server 113  
17 sends the resulting media content 115 back to mobile device  
18 105 in step 245. Media content 115 then causes the proper  
19 site to be displayed on the screen of mobile device 105 in  
20 step 247.

21 Turning next to FIG. 3A, shown is a schematic diagram  
22 depicting the product architecture of the ScanZoom  
23 software. The ScanZoom software is composed of three main  
24 modules: the application program interface ("API") 301,

1 decoding engine 303, and camera control module 305. API  
2 301 is the interface which the user utilizes to interact  
3 with the ScanZoom software. API 301 guides the user  
4 through all the steps required to decode and process  
5 barcode 103.

6 The steps utilized to acquire the barcode image and  
7 prepare it for decoding are shown in greater detail in FIG.  
8 3B. As shown, API 301 first directs camera control module  
9 305 to initialize the application ID for digital camera 107  
10 in step 351. This sets up digital camera 107 to accept  
11 input. Next, API 301 causes a "viewfinder" window to open  
12 on mobile device 105's display in step 353. This allows  
13 the user to make sure that barcode 103 is in proper focus  
14 and amply illuminated.

15 API 301 next processes the windows messages in step  
16 355. Windows messages are messages from the camera that  
17 interrupt the ScanZoom program if certain events have  
18 occurred. For example, a windows message may indicate that  
19 a new SMS message has been received or there is an incoming  
20 telephone call. Once the windows messages have been  
21 properly processed, API 301 checks to see if the windows  
22 message loop is empty in step 357. If the windows message  
23 loop is not empty, camera control module 305 closes digital  
24 camera 107s application ID in step 359. All message and

1 pictures are then cleared from the camera's active memory  
2 in step 361.

3 However, if the windows message loop is empty, the  
4 user is then directed to push a button on mobile device 105  
5 to take an image of barcode 103 in step 363.

6 Alternatively, the image may be acquired automatically by  
7 mobile device 105 for "on-the-fly" decoding. Camera  
8 control module 305 interacts directly with API 301 to  
9 perform this step since the user must physically push a  
10 button on mobile device 105 to take the image.

11 Alternatively, this feature may be fully automated.

12 Once the image of barcode 103 has been acquired and  
13 stored in memory, the image undergoes exposure control in  
14 step 365 to correctly compensate for the illumination.  
15 More specifically, this step correctly adjusts camera  
16 features such as the shutter speed. For example, if the  
17 ambient light is low, exposure compensation step 365 may  
18 compensate for this by increasing the shutter speed.

19 At this point in the image acquisition process, the  
20 acquired image is displayed to the user for review in step  
21 367. If the user chooses to accept the acquired image,  
22 camera control module 305 next converts the image to an  
23 eight-bit grayscale image in step 369. Camera control  
24 module 305 additionally removes artifacts from the image

1 which are typical of digital images. The artifacts are  
2 removed using pattern recognition because barcodes have  
3 definite shapes and alien features may be easily eliminated  
4 using pattern recognition.

5 The processed image is then passed to decoding engine  
6 303 for decoding. If the decoding is not successful, API  
7 301 returns the user to step 355 so that a new image can be  
8 acquired. However, if decoding is successful, API 301  
9 closes the camera in step 359 and clears the camera's  
10 memory in step 361.

11 Now referring back to FIG. 3A, decoding engine 303  
12 (utilized in step 205 of FIG. 2A) is responsible for  
13 decoding barcode 103 acquired via digital camera 107.  
14 Decoding engine 303 is designed to accommodate variations  
15 in brightness and contrast in the scanned image of barcode  
16 103. This is done through use of globally and locally  
17 adaptive image processing operations. Exposure levels can  
18 be very high or very low, without significant adverse  
19 affect on success of decoding. If contrast is low either  
20 because the ink presents little contrast with the  
21 substrate, or because the lighting conditions are poor,  
22 decoding engine 303 may still decipher barcode 103. Even  
23 highly variable shading within an image is recognized and  
24 compensated for. The underlying technique utilized by

1 decoding engine 303 to recognize features of barcode 103 is  
2 the detection of local pixel intensity patterns that may  
3 signal the presence of particular barcode features. This  
4 is in contrast to the approach of typical decode algorithms  
5 for more highly controlled commercial scanner or laser gun  
6 environments which typically do fixed thresholding or  
7 limited digital filtering which presumes a highly  
8 controlled environment and lighting configuration.

9       Decoding engine 303 is able to decode one and two  
10 dimensional barcodes with a CIF (typically 352x288) imager,  
11 and essentially all commonly used barcodes with a VGA  
12 (640x480) imager. Increasing the imager resolution  
13 generally improves the usability, decoding speed, and  
14 accuracy while increasing the range of viable barcodes.

15       In ordinary application usage, decoding engine 303  
16 does not require special illumination sources due to its  
17 ability to decode barcodes from images with low contrast.  
18 For color imagers, decoding engine 303 utilizes specific  
19 image processing algorithms in order to avoid problematic  
20 image artifacts during the translation from color to  
21 grayscale. Decoding engine 303 utilizes fast image  
22 processing algorithms to perform the conversion so that the  
23 maximum amount of information is preserved, making for a  
24 robust, easy to use reader.

1       Decoding engine 303 is also able to cope with moderate  
2 amounts of image focus global impairment due to distance  
3 and lens focal length issues. Additionally, the decoding  
4 algorithm is optimized to work reliably even with  
5 appropriate low-cost lenses in inexpensive consumer  
6 cameras.

7       Decoding engine 303 is also designed to perform  
8 reliably in difficult decoding situations. It is  
9 successful in variable light, low contrast, low resolution,  
10 focus, and other impaired conditions. These abilities make  
11 decoding engine 303 perfectly suited to capture and decode  
12 barcode images in a variety of "real world" embedded  
13 digital camera device conditions.

14       More specifically, key technical decoding features  
15 used in decoding engine 303 include:

16   **Rotation:**

17       Decoding engine 303 enables identification and  
18 decoding of most barcodes at any degree of rotation from  
19 the normal orientation. Decoding engine 303 is designed  
20 for the more general "any orientation" case.

21   **Geometric Distortions:**

22       Decoding engine 303 is tolerant of "aspect ratio,"  
23 "shear," "perspective," and other geometric image  
24 distortions. These distortions can be caused by a number of

1 things such as the camera line of focus not being  
2 perpendicular to the plane of the barcode. Specific  
3 algorithms can tolerate deviations from the perpendicular  
4 in any direction.

5 ***Adaptive Correction:***

6 One of the techniques used in several ways by decoding  
7 engine 303 is an adaptive, "multiple hypotheses" approach  
8 to detect the presence of specific features within barcode  
9 103. In general, while decoding an image of barcode 103, a  
10 number of assumptions are made by decoding engine 303 about  
11 how characteristic features of barcode 103 are likely to  
12 appear in an image. For example, the precise width and  
13 intensity of a minimal bar in an image and the threshold at  
14 which a data bit in a matrix code is counted as on or off  
15 are critical to decoding an image. Initial default  
16 estimates of these parameters may be wrong, and only by  
17 adaptively correcting them can the image be decoded. Where  
18 appropriate, decoding engine 303 will re-examine an image  
19 that has failed to decode under one set of assumptions and  
20 introduce revised assumptions to improve the likelihood of  
21 correctly decoding barcode 103.

22 ***Error Correction:***

23 Decoding engine 303 additionally makes use of  
24 sophisticated error correction technology for two-



1 dimensional barcode formats. The standard technique for  
2 error correction in dense barcodes is some variant of a  
3 "Reed-Solomon" algorithm. Decoding engine 303 uses the  
4 full power of this approach. Reed-Solomon techniques can  
5 correct a limited number of errors in these guesses.  
6 Decoding engine 303 makes guesses on most elements, but  
7 also identifies elements that are too poorly imaged or  
8 printed to make a reasonable guess. These are "erasures."  
9 Reed-Solomon error correction techniques can detect and  
10 correct more errors and thus has improved general results  
11 when erasures are identified.

12 ***Sub-Pixel Precision:***

13 Decoding engine 303 also allows barcode information to  
14 be resolved to sub-pixel precision. The algorithms need  
15 to, and can, with certain barcode types, retrieve  
16 information from a code element occupying an area less than  
17 1.5 X 1.5 pixels. Among the techniques employed by  
18 decoding engine 303 are specialized adaptive interpolation  
19 algorithms that take into account the precise local  
20 conditions surrounding the data feature being examined.  
21 Local conditions may include differences in lighting or  
22 printing quality, or secondary light scattering. Various  
23 image kernel operations are available to enhance the local  
24 image quality. The resulting outcome is better decoding

1 accuracy, support for higher density codes and more robust  
2 performance.

3         Decoding engine 303 may utilize any number of symbol  
4 libraries to resolve the correct barcode information. As  
5 shown, decoding engine 303 may access UPN-A/E library 307,  
6 RSS library 309, OI library 311, PDF417 library 313, QR  
7 code library 315, Code 39 library 317, Code 128 library  
8 319, EAN library 321, and JAN library 323.

9         Finally, camera control module 305 operates in  
10 conjunction with API 301 to enable a user to take a  
11 photograph of barcode 103 with digital camera 107.

12         Now referring to FIG. 4A, shown is a flowchart of the  
13 steps utilized by decoding engine 303 to enhance the image  
14 of barcode 103. Decoding engine 303 first attempts to de-  
15 skew the barcode image in step 401. Generally, skew occurs  
16 when the barcode picture is taken at an angle. To  
17 compensate for this effect, decoding engine 303 first  
18 identifies the angle(s) of skew in the image and processes  
19 the picture accordingly to remove the skew.

20         Next, decoding engine 303 attempts to repair images  
21 which exhibit yaw in step 403. Yaw occurs when the barcode  
22 or camera is moved during exposure, causing the image to  
23 exhibit streaks. Decoding engine 303 removes the yaw from

1 images by using a filter specifically designed to remove  
2 such effects.

3       Once the skew and yaw in the image has been corrected,  
4 decoding engine 303 attempts to remove any rotation of the  
5 barcode from the normal orientation which may have occurred  
6 during imaging. This may be accomplished in a variety of  
7 ways in step 405. For example, decoding engine 303 may  
8 first identify the angle of rotation of the image. This is  
9 much simpler for one-dimensional barcodes, but is also  
10 possible for two-dimensional barcodes. For one-dimensional  
11 barcodes, decoding engine 303 only has to calculate the  
12 angle at which the parallel bars in the barcode are rotated  
13 from the normal orientation. Once this has been  
14 determined, decoding engine 303 can apply a rotation  
15 function to the image to return the barcode image to the  
16 normal orientation.

17       Returning two-dimensional barcodes to a normal  
18 orientation requires much more processing because two-  
19 dimensional barcodes contain data in both the horizontal  
20 and vertical directions. To determine the angle of  
21 rotation, the barcode must be analyzed from at least two  
22 orientations, preferably perpendicular to each other. The  
23 results of the two analyzations can then be utilized to  
24 determine the angle of rotation of the two-dimensional

1 barcode. The same rotation function used for one-  
2 dimensional barcodes, previously described, can also be  
3 used for two-dimensional barcodes to return the barcode  
4 image to the normal orientation.

5       Next, decoding engine 303 sharpens the image using  
6 either a standard sharpening filter or a proprietary filter  
7 in step 406. The sharpening filter algorithm, described in  
8 FIG. 5A, has been shown to be effective for sharpening  
9 images containing barcodes. First, the sharpening  
10 algorithm converts the gray-scale barcode image is broken  
11 down into a two-dimensional array in step 501. Each entry  
12 in the two-dimensional array stores the horizontal and  
13 vertical coordinates (i.e., the "x" and "y" coordinates) of  
14 a single pixel. The image is then divided into an equal  
15 number of vertical sections in step 503. The number of  
16 sections ("ns") is equal to the width of the image (in  
17 pixels) divided by the desired width of the sections  
18 ("ws"). The width of the sections can either be user  
19 defined or automatically defined depending upon the size of  
20 the image. This converts the image to a three-dimensional  
21 array since each pixel also has an assigned section.

22       After the image has been divided into sections, the  
23 sharpening algorithm determines the minimum intensity of a  
24 pixel in each section in step 505. The image is then

1 processed linearly section by section in step 507. This is  
2 done by assigning a pixel intensity of zero to all pixel  
3 intensities which are below a threshold black level. The  
4 threshold black level is user-defined and may be changed  
5 for each image or section being processed. In contrast,  
6 all pixel intensities having a pixel value above a  
7 threshold white value are assigned a pixel intensity of  
8 255.

9 A pixel is also assigned a zero intensity if:

- 10 1. The value of the pixel lies within a predetermined  
11 range of the minimum pixel intensity for that section;  
12 or
- 13 2. The intensity of pixels surrounding a certain pixel  
14 has an intensity that lies within the predetermined  
15 range of minimum pixel intensity for that section.

16 After the image of the barcode has been processed in  
17 step 507, the sharpening algorithm renders the processed  
18 image sections back into an image. An example input and  
19 output barcode which have been processed by the  
20 aforementioned sharpening algorithm are shown in FIG. 5B  
21 and FIG. 5C, respectively. The outputted image of FIG. 5C  
22 has a much higher chance of being properly decoded than the  
23 inputted image of FIG. 5B.

1        Now referring back to FIG. 4A, decoding engine 303  
2   applies an edge enhancement filter to the image in step  
3   407. This further removes any image anomalies which may  
4   have occurred during imaging or conversion to black and  
5   white. Once the edges are enhanced, decoding engine 303  
6   counts the number of edges which occur in the barcode image  
7   in step 409. An edge is a point in the image where there  
8   is a sudden change in the color values of the image. An  
9   edge that defines a transition from white to black (light  
10   to dark) is called a rising edge and an edge that defines  
11   the transition from black to white (dark to light) is  
12   called a falling edge. Since the quality of the image  
13   returned by the camera of the cell phone isn't of a very  
14   good quality, the edge detection process relies on the  
15   series of approximations and sub processes. Thus the edge  
16   detection of step 409 returns a collection of edges (i.e.,  
17   points where it is believed that the value of the color  
18   changed from dark to light or light to dark).

19        If the number of detected edges is less than 25 as  
20   checked in step 411, decoding algorithm 303 attempts to  
21   adjust the barcode image again using a new set of  
22   assumptions in step 413. The image is then reprocessed  
23   using an unaltered version of the image stored in a buffer.  
24   If more than 25 edges are not detected after a number of

1 iterations, the ScanZoom application informs the user that  
2 a barcode could not be located and the application  
3 terminates.

4       However, if the number of edges is found to be greater  
5 than or equal to 25, decoding engine 303 advances to the  
6 flowchart of FIG. 4B. As shown in the flowchart, decoding  
7 engine 303 loads a first symbology library in step 451.  
8 The symbology library may be UPN-A/E library 307, RSS  
9 library 309, OI library 311, PDF417 library 313, QR code  
10 library 315, Code 39 library 317, Code 128 library 319, EAN  
11 library 321, and JAN library 323 (see FIG. 3A). Decoding  
12 engine then compares the number of edges a barcode needs to  
13 be in this library to the number of edges detected in the  
14 actual scanned barcode in step 453. If the number of edges  
15 does not match, decoding engine 303 loads the next  
16 symbology library in step 455 and repeats the edge  
17 comparison with the new library. Detection engine 303  
18 continues this comparison until a match is found.

19       When a match is found in step 453, detection engine  
20 303 proceeds to decode the start character of the barcode  
21 from the barcode image in step 457. Typically, the first  
22 character in a barcode indicates the use of the code. For  
23 example, in UPC codes, if the start character is a six, it  
24 indicates that it the barcode is a standard UPC code.

1 Detection engine 303 then decodes the middle characters of  
2 the barcode in step 459 and the stop character of the  
3 barcode in step 461.

4 Each barcode font relies on static tables of values to  
5 decode each character. These tables are designed using the  
6 bit patterns in the barcode font's specification. Each  
7 barcode consists of both dark and light elements. The  
8 combination of the dark and light elements and their  
9 thickness decides the value of the character. The  
10 difference of two consecutive edges provides the width of  
11 an element. The widths of the elements are then converted  
12 into bit patterns and mapped against the static tables to  
13 derive the character value. If a good value is returned  
14 then the next set of edges is passed. A bad value signifies  
15 that the barcode character could not be decoded.

16 In many barcode formats, the stop character is used as  
17 a check digit to ensure that the barcode has been decoded  
18 correctly. Detection engine 303 performs such an integrity  
19 check in step 463. If the integrity check fails, detection  
20 engine either loads another symbology library in step 455  
21 or attempts to re-decode the image in step 457. If the  
22 integrity check is passed in step 463, decoding engine 303  
23 is terminated in step 465 and the barcode data is forwarded  
24 to the messaging system (see FIG. 2A, step 209).



1       While the foregoing embodiments of the invention have  
2   been set forth in considerable detail for the purposes of  
3   making a complete disclosure, it should be evident to one  
4   skilled in the art that multiple changes may be made to the  
5   aforementioned description without departing from the  
6   spirit of the invention.